Code as Policies: Language Model Programs for Embodied Control

Year: 2022 | Citations: 1215 | Authors: Jacky Liang, Wenlong Huang, F. Xia

Abstract

Large language models (LLMs) trained on code-completion have been shown to be capable of synthesizing simple Python programs from docstrings [1]. We find that these code-writing LLMs can be re-purposed to write robot policy code, given natural language commands. Specifically, policy code can express functions or feedback loops that process perception outputs (e.g., from object detectors [2], [3]) and parameterize control primitive APIs. When provided as input several example language commands (formatted as comments) followed by corresponding policy code (via few-shot prompting), LLMs can take in new commands and autonomously re-compose API calls to generate new policy code respectively. By chaining classic logic structures and referencing third-party libraries (e.g., NumPy, Shapely) to perform arithmetic, LLMs used in this way can write robot policies that (i) exhibit spatial-geometric reasoning, (ii) generalize to new instructions, and (iii) prescribe precise values (e.g., velocities) to ambiguous descriptions ('faster') depending on context (i.e., behavioral commonsense). This paper presents Code as Policies: a robot-centric formulation of language model generated programs (LMPs) that can represent reactive policies (e.g., impedance controllers), as well as waypoint-based policies (vision-based pick and place, trajectory-based control), demonstrated across multiple real robot platforms. Central to our approach is prompting hierarchical code-gen (recursively defining undefined functions), which can write more complex code and also improves state-of-the-art to solve 39.8% of problems on the HumanEval [1] benchmark. Code and videos are available at https://code-as-policies.github.io